LSI DOCKET NO. 01-572

APPLICATION FOR
LETTERS PATENT OF THE UNITED STATES

<table>
<tr><td>

CERTIFICATE OF MAILING BY "EXPRESS MAIL"

"EXPRESS MAIL" Mailing Label Number EL750736962US

Date of Deposit: November 15, 2001

I HEREBY CERTIFY THAT THIS CORRESPONDENCE, **CONSISTING OF 14 PAGES OF SPECIFICATION AND 5 PAGES OF DRAWINGS**, IS BEING DEPOSITED WITH THE UNITED STATES POSTAL SERVICE "EXPRESS MAIL POST OFFICE TO ADDRESSEE" SERVICE UNDER 37 CFR 1.10 ON THE DATE INDICATED ABOVE AND IS ADDRESSED TO: BOX PATENT APPLICATION, THE COMMISSIONER OF PATENTS & TRADEMARKS, WASHINGTON D.C. 20231.

BY: _Lizzy Perkins._
      Lizzy Perkins

</td></tr>
</table>

SPECIFICATION

To all whom it may concern:

Be It Known, That I, **Alan D. Poeppelman, a citizen of the United States of America, residing at 4249 McMurray Avenue, Fort Collins, Colorado 80525**, has invented certain new and useful improvements in "**Method and Apparatus for Enhancing Correction Power of Reverse Order Error Correction Codes**", of which I declare the following to be a full, clear and exact description:

BACKGROUND OF THE INVENTION

1. Technical Field:

The present invention is directed generally toward a method and apparatus for enhancing the correction power of reverse order error correction codes.

2. Description of the Related Art:

Read channel integrated circuits for magnetic storage often contain significant digital signal processing and error correction logic to reliably reconstruct the original data stored on the media. Simplified block diagrams for this signal processing and error correction functions are given in **Figures 1A** and **1B** for a typical implementation of the encode and decode side of the process. Particularly, **Figure 1A** illustrates a typical implementation of an encode process. Data is received by error correction code (ECC) encoder **105**. ECC encoder **105** generates check bits, which are typically concatenated to the end the sector.. Thus, a predetermined number of bits of error correction code are computed for every unit of data being stored. When data is retrieved, the ECC syndromes are computed to determine if any of the data bits have been corrupted. Traditionally, the ECC encoder resides in the data controller chip.

Constraint encoder **110** receives the ECC encoded data from the ECC encoder. Constraint encoding is an encoding method commonly used on magnetic disks. A run length limited (RLL) encoder is an example of a constraint encoder. The actual number of bits recorded on the disk is greater than the data bits. The data is encoded to limit the length of certain patterns in the data for timing recovery and/or error suppression reasons. As electronics improve, fewer extra bits are inserted, and the ratio of data bits to recorded bits becomes greater. The "run length" is the number of consecutive zeros before a one bit is recorded. For example, RLL 1,7 means there must be at least one zero bit between every one bit, and the 7 means a maximum of eight time periods between flux transitions. Constraint **110** and the remainder of the encode process typically resides in the read channel chip. The constraint encoded data is received by post processing encoder **115**, which further encodes data for write logic and driver **120**. The write logic and driver outputs analog data to the storage device.

Turning to **Figure 1B**, a typical decode process is depicted. Analog front end (AFE) **150** receives analog data in from a storage device and analog-to-digital converter (ADC) **155** converts the data to digital. Finite impulse response (FIR) **160** filters the data. Viterbi detector **165** is a maximum likelihood detector that computes the most likely data sequence. Post processor **170**

5   performs processing on the Viterbi data. Constraint decoder **175** removes the constraint code. These elements of the decode process traditionally reside in the read channel chip. ECC decoder **180** decodes the error correction code and outputs corrected data. The ECC decoder traditionally resides in the data controller chip.

Analog Front End accepts analog signals from the pre-amp and filters it before the ADC

10   digitizes it. The FIR (Finite Impulse Response) Filter conditions the digitized signal by digitally filtering it and attempts to match a "target".

As silicon processing improved over time allowing increased integration of functions on the same chip and as the error propagation properties of the constraint decoder have worsened, the idea of reverse order ECC has emerged. The basic idea of reverse order ECC, sometimes

15   called reverse ECC, is to switch the order of the constraint and ECC encoding/decoding. Switching the order of the constraint decoder and the ECC decoder allows the full strength of the ECC to be utilized without the diluting effect of the constraint decoder's error multiplication properties. This change in structure for reverse ECC is shown in **Figures 2A** and **2B**.

Particularly, **Figure 2A** illustrates a typical reverse ECC encode process. In the encode

20   process of **Figure 2A**, constraint encoder **205** and ECC encoder **210** are reversed. The ECC bits, which are typically concatenated at the end of the data sector, may not conform to the constraint requirements and may need to pass through a separate and different constraint encoder. Post processing encoder **215** may not be beneficial and may be removed from the process. Turning to **Figure 2B**, a typical reverse ECC decode process is depicted. In the decode process of **Figure**

25   **2B**, ECC decoder **275** and constraint decoder **280** are reversed. Thus, the full strength of ECC decoder **275** may be realized without errors being multiplied by constraint decoder **280**.

A Viterbi detector finds the most likely binary sequence given a stream of digitized data in a least mean square error sense. A Viterbi detector is used when inter-symbol interference is involved in the signal, that is, when signals overlap each other.

LSI DOCKET NO. 01-572

If the encoding of a constraint code spans over multiple ECC symbols, error propagation can occur. This is because an error might be limited to a short span before constraint code decoding but fans out to additional bits when decoded. An increase in errors, due to errors being multiplied by a constraint decoder, results in a higher sector retry/failure rate. A retry due to

5   uncorrectable data is time consuming. Therefore, it would be advantageous to provide improved correction power for ECC decoding during a reverse ECC decode process to avoid an increase in the number of repeated retrievals due to uncorrectable data.

SUMMARY OF THE INVENTION

The present invention takes advantage of information available from a post processor. This information is a list of highly probable error event patterns and locations found by employing a list

5   Viterbi or a set of matched filters on Viterbi data. This list of possible errors can be used by the ECC decoder in an iterative process whenever the correction power of the ECC is exceeded. If the ECC decoder cannot correct the data on its first unassisted try, an iterative process is employed which, in essence, modifies the data with potential errors identified from the list created by the post processor and tries the correction process over again. An algorithm may be employed to try each

10   error singly or in combination with other errors. This iterative process continues until a correctable indication is given by the ECC algorithm. The data is then corrected with the ECC results and the corresponding error list.

BRIEF DESCRIPTION OF THE DRAWINGS

The novel features believed characteristic of the invention are set forth in the appended claims. The invention itself however, as well as a preferred mode of use, further objects and advantages thereof, will best be understood by reference to the following detailed description of an illustrative embodiment when read in conjunction with the accompanying drawings, wherein:

Figures 1A and **1B** depict a prior art implementation of encode and decode processes;

Figures 2A and **2B** illustrate prior art reverse ECC encode and decode processes;

Figure 3 is an iterative ECC decode process in accordance with a preferred embodiment of the present invention;

Figure 4 is an iterative ECC decode process employing a small ECC decoder in accordance with a preferred embodiment of the present invention; and

Figure 5 is a flowchart of the operation of the ECC decoder process in accordance with a preferred embodiment of the present invention.

DETAILED DESCRIPTION

The description of the preferred embodiment of the present invention has been presented for purposes of illustration and description, but is not limited to be exhaustive or limited to the

5   invention in the form disclosed. Many modifications and variations will be apparent to those of ordinary skill in the art. The embodiment was chosen and described in order to best explain the principles of the invention the practical application to enable others of ordinary skill in the art to understand the invention for various embodiments with various modifications as are suited to the particular use contemplated.

10   With reference now to the figures and in particular with reference to **Figure 3**, an iterative ECC decode process is shown in accordance with a preferred embodiment of the present invention. Data equalized to the target is received by Viterbi detector **302** and matched filters **304**. The Viterbi detector provides data to ECC syndrome generator **306**. ECC decoder **320** attempts to correct the data given the ECC syndromes of data. If the data is correctable **322**,

15   ECC corrector and error event corrector **324** produces corrected data out. The Viterbi detector produces "ones and zeroes" from the digitized data stream output of the FIR. The ECC decoder provides a list of locations and bits to toggle to the ECC corrector and event corrector.

One enhancement to the invention is to calculate the ECC syndromes for each error event supplied by the post processor and for the raw data itself. By making use of the linear property of

20   ECC codes, the ECC syndrome for iterative trials may be created simply by performing an exclusive-OR (XOR) on the ECC syndromes of data with all the ECC syndromes of errors used in the trial. This saves the computational effort of recalculating the ECC syndromes for each trial by the polynomial division method.

Thus, matched filters **304** produce a list of most likely errors and metrics **308** given the

25   data equalized to the target and the ECC syndromes of data. The list of most likely errors and metrics produces ECC syndromes of errors **310**. Syndrome combiner **312** combines the ECC syndromes of data and the ECC syndromes of errors. The ECC decoder may then try combinations of syndromes until a correctable indication is given in **322**.

Each trial would XOR the data syndromes with the corresponding error syndromes. If the trial was for error number 1 and 3 on the list, the 3 sets of syndromes would be XORed together -- data, error 1, and error 3. The ECC decoder algorithm has a predetermined number of ECC symbols it can correct. A symbol typically is 8 or 10 bits. The combiner combines syndromes

5   until a correctable result is achieved or until all trials are exhausted. If a correction location is beyond the end of the sector or if there are fewer roots found to the "error locator polynomial" than the degree of the polynomial, then the data is uncorrectable. The corrected data out is passed to the constraint decoder.

With reference now to **Figure 4**, an iterative ECC decode process employing a small

10   ECC decoder is shown in accordance with a preferred embodiment of the present invention. The process of **Figure 4** is similar to the process of **Figure 3**. Mini ECC decoder **430** is employed to correct sectors with only a few errors, which occur frequently. Mini ECC decoder **430** receives ECC syndromes of data and the ECC algorithm determines whether the data is correctable **432**. If the data is correctable in **432**, ECC corrector and error event corrector **424** produces corrected

15   data out.

If, however, the data is not correctable in **432**, full ECC decoder **420** corrects the sectors with more errors, which occur much less frequently. The full ECC decoder could operate in either the single unassisted mode or in the assisted iterative mode. The mini ECC decoder would use a subset of the partial syndromes generated for the full ECC decoder so no additional

20   redundancy bits are required. The mini ECC decoder would defer to the full ECC whenever an uncorrectable condition is flagged by the mini ECC correction algorithm.

With reference to **Figure 5**, a flowchart of the operation of the ECC decoder process is shown in accordance with a preferred embodiment of the present invention. The process begins and receives Viterbi data and most likely errors and metrics (step **502**). The process then

25   calculates ECC syndromes of data (step **504**) and a subset of the partial syndromes is used by the mini ECC to offload the full ECC (step **506**). Next, a determination is made as to whether the data is correctable by the mini-ECC (step **508**). If correctable by the mini-ECC, the data is corrected (step **510**) and the process ends.

LSI DOCKET NO. 01-572

The process also calculates ECC syndromes of errors (step **512**). If the data is not correctable in step **508**, the full set of syndromes is used by the full ECC decoder (step **514**) and a determination is made as to whether the data is correctable (step **516**). If the data is not correctable, the process combines syndromes (step **518**) and the combined syndromes are used in

5   the full ECC decoder (step **520**). Then, the process returns to step **516** to determine whether the data is correctable. If the data is correctable in step **516**, the process proceeds to step **510** to correct the data. Thereafter, the process ends.

Thus, the present invention solves the disadvantages of the prior art by providing an ECC decoder algorithm that takes advantage of information available from a post processor. This

10   information is a list of highly probable error event patterns and locations found by employing a list Viterbi or a set of matched filters on Viterbi data. This list of possible errors can be used by the ECC decoder in an iterative process whenever the correction power of the ECC is exceeded. If the ECC decoder cannot correct the data on its first unassisted try, an iterative process is employed which, in essence, modifies the data with potential errors identified from the list created by the post

15   processor and tries the correction process over again. An algorithm may be employed to try each error singly or in combination with other errors. This iterative process continues until a correctable indication is given by the ECC algorithm. The data is then corrected with the ECC results and the corresponding error list. The invention has the advantage of higher ECC correction power without the corresponding increase in redundancy bits. This allows for higher data coding efficiency on the

20   media. Conversely, the invention allows for a lower sector retry/failure rate due to the increase in correction power and lower uncorrectable sector rate.